



# TC update

Amir Vadai, Or Gerlitz, Rony Efraim

Netdev 2.1, Montreal, April 2017

# use cases for packet Headers re-write with TC

- Flow based Routing for virtual environments (e.g Open-Stack DVR)
  - re-write L2 headers with the routers' mac and decrement TTL during virtual switch/router operations
- Flow based Stateless NAT
  - re-write L3/L4 headers by virtual switch
  - can be in parallel with routing
- Flow based termination/modifications of TCP connections
  - re-write sequence numbers
- more? sure, you say

# TC action for packet Header re-write

- the action is called **pedit** (Packet Edit)
- the pedit action contains set of keys each defining a re-write element
- each key has <offset, mask, value> and few more legacy fields (not explained here), all are 32 bits in size
- <offset, mask, value> define byte offset into the packet where the masked value is written (potentially with manipulations)
- offset 0 assumed to be where an IP header starts
- the kernel UAPI is RAW in the sense that no structuring is assumed on packets

# towards HW offloading of header re-write with TC

- decided not to re-invent the wheel, re-use pedit (be environmental, even if Mr. Trump doesn't think it's needed)
- kept the <offset, mask, value> triplet
- added notion of **header type** to where the offset applies
  - header type follows the kernel `skb_mac/network/transport/_header` concept
  - but with a little twist that states the network (IPV4/V6) and transport (TCP/UDP)
- the kernel UAPI remained RAW in the sense that explicit fields (e.g mac, ttl, port) are not modeled, and the user has to specify their offset in the relevant header
- added notion of **command**, currently supporting set or add
- the add command uses unsigned integer arithmetic (add 255 → dec TTL by 1)
- work by Amir Vadai, merged in 4.11

# example: re-write src/dst mac and decrement TTL

```
$ tc filter add dev enp1s0 protocol ip parent ffff: prio 100 flower skip_sw ip_proto udp dst_port 7000 action pedit munge eth dst set 11:22:33:44:55:66 pedit munge eth src set aa:bb:cc:dd:ee:ff pedit munge ip ttl add 255
```

```
$ tc filter show dev enp1s0 protocol ip parent ffff:
filter pref 100 flower
filter pref 100 flower handle 0x1
  eth_type ipv4
  ip_proto udp
  dst_port 7000
  skip_sw
    action order 1: pedit action pass keys 5
    index 69 ref 1 bind 1
    key #0 at eth+0: val 11223344 mask 00000000 (re the polarity of masks, ask Jamal...)
    key #1 at eth+4: val 55660000 mask 0000ffff
    key #2 at eth+4: val 0000aabb mask ffff0000
    key #3 at eth+8: val ccddeeff mask 00000000
    key #4 at ipv4+8: add ff000000 mask 00ffffff
```

# HW offloading of pedit with ConnectX5

- the driver translates the per key {command, header-type, <offset, mask, val>} into HW modify header action descriptor
- The translation logic uses the header type + offset to realize what field (e.g ETH DMAC, IP TTL, UDP port) out of the HW parser enumeration to write.
- The mask potentially further defines offset and length within the field
- the derived set of descriptors is represented by modify header ID which is then attached as an action to the steering specification of the relevant flow
- Header re-write is supported for SRIOV e-switch steering and for NIC RX steering
- work by Or G., accepted for 4.12



# Thank You